ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

на тему:

Построение правил синтаксического анализа запросов пользователя в вопросно-ответной системе

основная образовательная программа бакалавриата по направлению подготовки 45.03.02 «Лингвистика»

Исполнитель:

Обучающийся 4 курса
Образовательной программы
«Прикладная, экспериментальная и
математическая лингвистика (английский язык)»
Профиль «Прикладная, экспериментальная и
математическая лингвистика»

очной формы обучения
Коваленко Владислав Сергеевич

Научный руководитель:
к.ф.н., доц. Митренина О.В.

Рецензент:
к.ф.н., ст. преп. Добров А.В.

Санкт-Петербург
2018

Введение

Проблема взаимодействия пользователя и компьютера возникла в середине XX в., когда операции на ЭВМ могли выполнять лишь обученные специалисты. Со временем порог, который необходимо перешагнуть пользователю компьютера, начал постепенно снижаться. С появлением простых операционных систем, которые затем были снабжены графическим пользовательским интерфейсом, все больше людей смогли выполнять работу на вычислительных машинах.

Однако на данный момент существует множество программ и приложений, для которых идеальным способом взаимодействия было бы общение с компьютером на естественном языке. Разработки в этом направлении ведутся еще с 60-х гг. прошлого века, но проблема далеко не решена.

Пользователь компьютера сегодня для выполнения все большего количества функций обращается к специализированным диалоговым системам: чат-ботам и вопросно-ответным системам. Для их функционирования крайне важно правильно обработать запрос пользователя. Анализ запроса проходит в несколько этапов, на каждом из которых парсер передает на следующий этап наиболее вероятные варианты разбора. Поскольку инструменты анализа на разных уровнях языка еще не идеальны (особенно для русского языка), верный вариант разбора может быть утерян на одном из ранних уровней, после чего парсер уже не сможет правильно проанализировать запрос.

Парсер Helis, разрабатываемый А.М. Поповым и Е.В. Еникеевой с 2017 года, обладает структурой и алгоритмом поиска, позволяющими обрабатывать и хранить все варианты синтаксического разбора до того, как они будут переданы на следующий уровень извлечения семантики (фактов).

Целью этой работы является написание правил синтаксического анализа в грамматике парсера Helis для обработки запросов пользователя на естественном языке, обращенных новостному чат-боту.

Поставленная цель предполагает выполнение следующих **задач**:

- изучение устройства диалоговых систем и их подвидов;
- обзор имеющихся способов извлечения семантики из запросов пользователей, в том числе на русском языке;
- изучение устройства парсера Helis, используемой им грамматики и структуры правил синтаксического анализа;
- написание правил синтаксического анализа:
    - исследование характера общения между человеком и чат-ботом;
    - поиск характерных синтаксических структур в запросах.

Материалом для исследования послужила выборка из 80 потенциальных запросов новостному чат-боту, собранная путем опроса респондентов.

Актуальность данной работы обусловлена тем, что большинство парсеров, занимающихся извлечением фактов из текстов на русском языке, основаны на правилах и не обладают высокой точностью работы.

Практическая значимость работы связана с возможностью использования результатов для создания новостного чат-бота, использующего парсер Helis для обработки запросов.

Глава 1 представляет собой обзор диалоговых систем и их подвидов, истории их развития и проблем, стоящих перед ними сейчас.

Глава 2 рассматривает проблему извлечения информации из текста и описывает, как этот метод применяется для выявления семантики текста, в том числе и в системах, работающих с русским языком. Также детально описан парсер Helis: принцип его работы, преимущества и отличия от других систем.

Глава 3 посвящена собственно проблеме написания правил синтаксического анализа. Представлены имеющиеся исследования на тему характера взаимодействия пользователя и узкоспециализированного диалогового агента. Описаны и кратко обоснованы полученные правила.

Благодарю А.М. Попова за научное консультирование, оказанное им в ходе написания данной работы, и за предоставление технической помощи, связанной с парсером Helis.

Глава 1. Обзор диалоговых систем

1.1. Диалоговые системы

Создание интерфейса для общения пользователя и машины является задачей, поставленной еще на заре развития компьютерных систем – в 60-х гг. прошлого века. Вычислительные способности компьютера позволяют использовать его для разных прикладных задач, будь то доступ к базам данных, выдача справки, ответ на вопрос или рекомендация в конкретной области знаний. Системы, направленные на поддержание подобного диалога на естественном языке – **диалоговые системы**, – позволяют заметно облегчить работу пользователя.

Однако вопрос о диалоге человека и машины поднимался и ранее: опубликованная в 1950 г. работа английского математика Алана Тьюринга «Вычислительные машины и разум» [Turing 1950] связывала способность вести диалог со способностью думать. Тьюринг заменил вопрос «Могут ли машины думать?» на «Могут ли машины выиграть “игру-имитацию”», то есть могут ли они при диалоге убедить собеседника, что он разговаривает с человеком. Этот философский вопрос был оформлен в виде так называемого теста Тьюринга, который задал направление развития диалоговых систем на многие годы вперед.

Так, одной из первых диалоговых систем, целью которых было прохождение теста Тьюринга, стала ELIZA – разработка Джозефа Вейценбаума, созданная в 1966 г. [Weizenbaum 1966]. Принцип её работы прост: ELIZA использует определенный сценарий, в рамках которого она симулирует психотерапевта. Система находила во фразах пользователя определенные ключевые слова, которые использовала для генерирования вопросов, якобы направленных на рефлексию собеседника. ELIZA не имеет никакого представления о смысловом содержании поступающих сообщений, однако многие люди, пообщавшись с таким компьютерным психотерапевтом, оказались убеждены в том, что система из понимает. Среди этих людей была и секретарша Вейценбаума [Pinker 1995]. Считается, что

ELIZA и схожие с ней программы действительно обладают потенциалом помочь людям, страдающим от психологических проблем [Colby et al. 1966], а версия системы даже была интегрирована в текстовый редактор GNU Emacs.

ELIZA считается одним из первых **чат-ботов** – систем, созданных для длительного общения, которые имитируют диалоговые черты человеческого общения [Jurafski, Martin 2017]. Существуют также чат-боты, направленные на выполнение определенных задач (task-oriented), как то покупка билетов на самолет или заполнение юридических документов. В эту категорию можно зачислить и **персональных помощников**. Такие голосовые системы как Siri (компании Apple), Cortana (Microsoft), Alexa (Amazon) способны выполнять множество задач. И хотя они имитируют человеческий разговорный стиль общения, они не предназначены для длительного диалога на определенную тему, а ранние версии этих систем и вовсе не были рассчитаны на последовательные запросы пользователя.

Существуют и **вопросно-ответные системы**, целью которых является развернутый ответ на поставленный вопрос [Jurafski, Martin 2017]. Это справочные системы, которые используют онтологии и огромные базы данных, корпуса текстов и веб-страниц.

Одной из самых известных систем подобного рода является IBM Watson, разработка которого ведется с 2005 г. Изначальной целью создателей было научить её побеждать в телеигре Jeopardy!, где игроки соревнуются друг с другом в ответе на вопросы из разных областей знаний. Чтобы соревноваться с людьми, Watson должен был уметь проанализировать вопрос и дать ответ за несколько секунд. Спустя 5 лет работы компьютер был способен обыгрывать людей, а в 2011 г. поучаствовал на самом телешоу, где обыграл двух чемпионов. К этому моменту Watson имел доступ к 4 терабайтам текстовых данных, среди которых осуществлялся быстрый поиск информации.

1.2. Основные принципы работы диалоговых систем

Работу диалоговых систем можно разбить на три основных элемента: *понимание естественного языка, диалоговый менеджер* и *генерация естественного языка* [Константинов, Дегтева 2016].

Первый элемент отвечает за получение семантического представления входного текста, которое используется в дальнейшей работе.

Диалоговый менеджер – ядро системы, которое следит за контекстом диалога, обрабатывает полученную информацию, решает, каково будет содержание следующего высказывания. Этот модуль является связующим звеном между модулями понимания и генерации языка.

Последний элемент автоматически создает высказывание на естественном языке, основываясь на информации, полученной из диалогового менеджера. Этот процесс делится на три стадии: *планирование документа, микропланирование* и *поверхностная реализация*. На первом этапе определяется и упорядочивается информация, которая лежит в основе выходного текста. На этапе микропланирования информация делится на условные предложения, выбираются подходящие слова для её выражения, генерируются отношения референции в тексте. На последнем этапе собственно создаются грамматически правильные предложения.

Таким образом система преобразует текст пользователя в понятную для неё структуру и анализирует её путем применения неких правил, которые определяют реакцию системы – ответ на естественном языке или же внеязыковое действие.

1.3. Основные проблемы диалоговых систем

Упомянутая выше система ELIZA с успехом заставляла людей считать, что компьютер понимает их слова, хотя это совсем не так. ELIZA использует популярный среди чат-ботов метод построения ответа на основе ключевых слов из запроса. Система пытается найти в речи собеседника слово, которое

бы имелось в её базе данных. Тогда она строит ответ, имитируя манеру психотерапевта, а именно – задает наводящий вопрос, поставив в его ядро найденное ключевое слово. Например [Weizenbaum 1966]:

User: You are not very aggressive but I think you don't want me to notice that.

ELIZA: WHAT MAKES YOU THINK I AM NOT VERY AGGRESSIVE

User: You don't argue with me.

ELIZA: WHY DO YOU THINK I DON'T ARGUE WITH YOU

В случае, если чат-боту не удается найти ключевых слов, то он отвечает одной из подготовленных фраз, которые просят пользователя выразить свою мысль другим образом. Например, "*Can you elaborate on that?*" или "*I'm not sure I understand you fully*".

Такой ограниченный набор средств приводит к основной проблеме подобных систем: они неправдоподобны в общении, поскольку даже не пытаются полностью понять смысл высказывания. Именно поэтому разработчики систем, участвующих в тесте Тьюринга, прорабатывают легенды для своих чат-ботов, которые бы оправдывали подобную манеру общения. Так, в 2014 году тест был пройден системой, которая отыгрывает роль 13-летнего мальчика из Украины, что объясняет грамматические неточности в ответах системы и её недостаточные знания о мире [Warwick, Shah 2016].

Другой подход к решению этой проблемы представлен в системе Cleverbot. Этот чат-бот основывается на данных об уже произошедших и сохраненных в памяти диалогах. Вместо того, чтобы выстраивать ответ при помощи заранее прописанных шаблонов, система находит в своей базе данных, как другие люди отвечали на подобный запрос. Благодаря накопленным данным и процессу самообучения Cleverbot более похож на человека в общении, чем другие чат-боты [Hill et al. 2015], но он всё еще не углубляется в семантику запросов.

Схожий метод используется в сервисе Talk to Books компании Google. Talk to Books отвечает на запрос пользователя цитатами из книг, которые наиболее вероятны последовать за запросом. В основе системы лежит алгоритм, обученный на корпусе из более чем 100.000 книг. Обучение производилось без учителя и заключалось в нахождении фраз, встречающихся вместе в одном и том же участке исходных данных. В результате система вывела 512 параметров, представляющих синтаксис и семантику естественного языка, благодаря которым находятся нужные ответы на входные данные [Cer et al. 2018].

IBM Watson, обладающий непревзойденными ресурсами, проводит более тщательный анализ запроса. В частности, на первом этапе работы системы происходит синтаксический парсинг, выявление кореференции, а также выделение именованных сущностей и отношений между ними (см. 2.1.). Помимо этого, обрабатывается и сам вопрос: система находит его область, тип требуемого ответа и фокус – ту часть вопроса, вместо которой можно было бы подставить искомый ответ [Jurafski, Martin 2017].

Работа Watson была заточена под определенный тип запросов – вопросы телеигры Jeopardy!, построенные по конкретной структуре. Однако система все равно способна на гораздо более глубокий уровень семантического анализа, чем чат-боты, описанные выше.

1.4. Выводы из главы 1

В этой главе мы рассмотрели несколько типов *диалоговых систем* – систем, осуществляющих диалог между машиной и пользователем на естественном языке. Выделяют *чат-ботов*, созданных для длительного общения; *персональных помощников*, предназначенных для непродолжительного общения, где ответом на запрос пользователя может являться внеязыковое действие; *вопросно-ответные системы*, которые способны дать развернутую справку на запрос пользователя.

Работу всех диалоговых систем можно разделить на три этапа: *понимание естественного языка, диалоговый менеджер* и *генерация естественного языка*. На первом этапе запрос пользователя преобразуется в понятную для системы структуру, на втором – система обрабатывает запрос при помощи правил и определяет реакцию на запрос, на третьем – система либо создает ответ на естественном языке, либо производит внеязыковое действие.

Была рассмотрена одна из важнейших проблем диалоговых систем – ограниченный семантический анализ. Так, *ELIZA* – один из первых успешных чат-ботов – лишь создает иллюзию того, что речь пользователя понимается, выделяя в ней ключевые слова, на основе которых и выстраивается ответ по заранее прописанным шаблонам. Этот принцип работы лежит в основе и большинства современных систем. Исключением является *Cleverbot*, который сохраняет историю переписки с каждым пользователем. Это позволяет системе отказаться от шаблонов: она выделяет ключевые слова, а затем находит в своей базе данных наиболее подходящий ответ. Более глубокий анализ проводит *IBM Watson*. Эта система обучена анализу вопросов из телеигры Jeopardy! и обладает обширными технологическими ресурсами, которые доступны немногим.

Глава 2. Связь синтаксического анализа с извлечением фактов из текста

2.1. Извлечение информации как метод представления семантики текста

Извлечение информации (IE, Information Extraction) – задача выявления структуры в документах, где готовой структуры нет. Из текста на естественном языке требуется получить однозначное логическое представление определенных событий и отношений в документе, которое уже поддается стандартным методам обработки [Jurafski, Martin 2017].

В этой области существует несколько подзадач. Наиболее простой из них является *распознавание именованных сущностей* (NER, Named Entity Recognition). **Именованные сущности** – слова в тексте, обозначающие конкретный предмет или явление, выделяющие конкретный предмет или явления из ряда однотипных. Обычно это имена собственные, которые обозначают людей, места и организации, но на практике это понятие расширяют так, что туда также входят обозначения дат и, например, цен. Задача NER заключается в нахождении в тексте именованных сущностей и обозначения их типа в специальной разметке.

Следующим этапом происходит *извлечение отношений* между найденными сущностями. Не стоит путать выявляемые отношения с семантическими отношениями между словами (синонимии, гипонимии, меронимии и др.). Рассматриваются бинарные отношения между сущностями, обозначаемыми словами в тексте, вроде «работает в», «является ребенком», «находится в». Эту подзадачу также иногда называют *извлечением фактов* или *событий*, где под фактами имеется в виду зафиксированное в тексте знание об объектах и их свойствах и отношениях между ними.

Описанная в главе 1 система IBM Watson выполняет извлечение именованных сущностей и отношений между ними. Рассмотрим её работу на следующем примере, приведенном в [Lally, Fodor 2011].

Системе дан следующий вопрос: *Poets and Poetry: He was a bank clerk in the Yukon before he published "Songs of a Sourdough" in 1907.* Она

определяет *Yukon* как геополитическую сущность (тег geopolitical entity), *“Songs of a Sourdough”* как произведение (тег composition) и производит кореференцию, связывая *he* и *clerk*. Затем извлекаются следующие отношения:

authorof (focus, “Songs of a sourdough”)

publish (e1, he, “Songs of a sourdough”)

in (e2, e1, 1907)

temporallink (publish(...), 1907)

Многие описываемые в текстах события соотносятся с типичными ситуациями в мире. Шенк и Абельсон ввели понятие *сценариев*, описывающих определенные типичные ситуации, назначая сущностям роли и устанавливая отношения между ними [Schank, Abelson 1977]. Сценарии можно представить при помощи *шаблонов* с заранее прописанными слотами, которые можно заполнить материалом из текста. Заполнение таких шаблонов при помощи информации, явно выраженной в тексте и логически выведенной из него, также является одной из подзадач.

Зачастую извлечение фактов представляется именно при помощи шаблонов или фреймов для конкретной ситуации. Например, для предложения «Яндекс купил Кинопоиск за $80 млн. в октябре 2013 г.» может быть построен следующий шаблон, описывающий ситуацию покупки:

Сумма	Покупатель	Объект	Продавец
$80 млн.	Яндекс	Кинопоиск	?

Таблица 1.

Задача извлечения информации тесно связана с *информационным поиском* (IR, Information Retrieval) и *обработкой естественного языка* (NLP, Natural Language Processing). Некоторые пакеты ПО, созданные для работы в области NLP, такие как NLTK, имеют модуль для выполнения задачи извлечения информации [Bird et al. 2009]. Есть и специализированные инструменты для выполнения этой задачи на материале русского языка, как Томита-парсер. Проводятся соревновательные конференции, где

разработчики работают в пределах заранее обозначенных тем. Это, например, серии конференций MUC (Message Understanding Conference) и ACE (Automatic Content Extraction), а также соревнование FactRuEval, которое было проведено на материалах русского языка в рамках конференции Диалог 2016. В частности, на ней рассматривались три дорожки: извлечение именованных сущностей, извлечение уникальных сущностей с заполнением их атрибутов и извлечение фактов. Наихудший результат участники показали на третьей дорожке. В ней участвовали лишь двое участников, и F1-мера победителя была равна 66%. Соответствующее значение для первых двух дорожек равно 93%. Такой сильный разрыв показывает, что на данный момент инструменты для извлечения информации способны уверенно решать лишь самые простые из подзадач [Старостин 2016].

2.2. Проблемы, связанные с синтаксическим анализом

Можно выделить два основных направления разработок в области извлечения информации: методы на основе правил и статистические методы. В первом случае разработчики строят лексико-синтаксические шаблоны, которые применяются к синтаксически обработанному тексту. Этот метод был предложен Марти Хёрст в 1992 году. Она дает следующий пример шаблона:

NP_H such as {NP,}* {(or|and)} NP,

где NP_H – гипероним для последующих NP, а в фигурных скобках заключены факультативные элементы.

В предложении “...works by such authors as Herrick, Goldsmith, and Shakespeare.” подобный шаблон выделит следующие три отношения:

hyponym(“author”, “Herrick”),

hyponym(“author”, “Goldsmith”),

hyponym(“author”, “Shakespeare”) [Hearst 1992].

Подход на основе правил, написанных вручную отличается высокой точностью работы, но среди его минусов высокая трудозатратность (поэтому подобные разработки ограничиваются лишь определенными областями знаний, где можно построить более конкретные и точные правила) и низкий уровень полноты. Большинство систем, работающих с русским языком, основаны на этом подходе [Старостин 2016].

Одним из факторов, влияющих на низкую полноту, является тот факт, что большинство синтаксических парсеров использует статистические методы и эвристики и выдает лишь самый вероятный вариант разбора предложения (в случае одноцелевых парсеров) или же несколько самых вероятных (в случае многоцелевых парсеров), чтобы избежать комбинаторного взрыва [Добров 2016]. Вполне возможно, что парсер таким образом отсекает верный вариант разбора, не передав его модулю IE.

В [Popov, Enikeeva 2017] авторы постулируют, что для улучшения работы IE-модуля требуется синтаксический парсер, который бы мог передать все варианты разбора, сжатые в специальную структуру, и особый алгоритм поиска, который бы работал с этой структурой. Для целей этой работы важен сам разрабатываемый ими парсер Helis.

2.3. Описание парсера Helis

2.3.1. Алгоритм CYK

Helis – парсер, который разрабатывается А.М. Поповым и Е.В. Еникеевой с 2017 года. Парсер написан на языке JavaScript и находится на ранней степени разработки. На момент написания работы в открытом доступе доступна версия программы, позволяющая проводить морфологический и синтаксический анализ текста на русском языке.

В основе парсера лежит алгоритм Кока-Янгера-Касами (Cocke-Younger-Kasami algorithm, CYK) [Kasami 1965, Younger 1967], который применяется к *контекстно-свободным грамматикам*. Они состоят из множеств терминалов и нетерминалов, начального символа грамматики и

правил вывода, где слева находятся одиночные нетерминалы, а справа ряд нетерминалов и терминалов. Важно, чтобы грамматика была представлена в нормальной форме Хомского, т.е. все её правила вывода имели следующий вид [Sipser 1997: 107]:

- $A \rightarrow a$, где A – терминал, и a – нетерминал;
- $A \rightarrow BC$, где B и C – неначальные нетерминалы;
- $S \rightarrow \varepsilon$, где S – начальный символ, а ε – пустая строка.

Поскольку любую контекстно-свободную грамматику можно привести к нормальной форме Хомского, поэтому CYK-алгоритм является универсальным.

Алгоритм позволяет узнать, выводимо ли некое слово в данной грамматике. Допустим, что имеется входное слово w длины n и контекстно-свободная грамматика Γ в нормальной форме Хомского с количеством нетерминалов N. Строится трехмерный массив $d[N, n, n]$, каждая ячейка которого будет заполняться логическими значениями на протяжении работы алгоритма. Ячейка $d[A, i, j]$ получит истинное значение только в том случае, если подстрока входного слова $w[i...j]$ выводима из нетерминала A. Перейдя к подстрокам длины больше 1, алгоритм начинает рассматривать все бинарные деления входного слова и ищет в грамматике такое правило вывода $A \rightarrow BC$, что левое деление выводимо из B, а правое – из C (Рис. 1). После окончании работы алгоритма в ячейке $d[S, 1, n]$ заключается ответ на вопрос, выводимо ли w из данной грамматики.

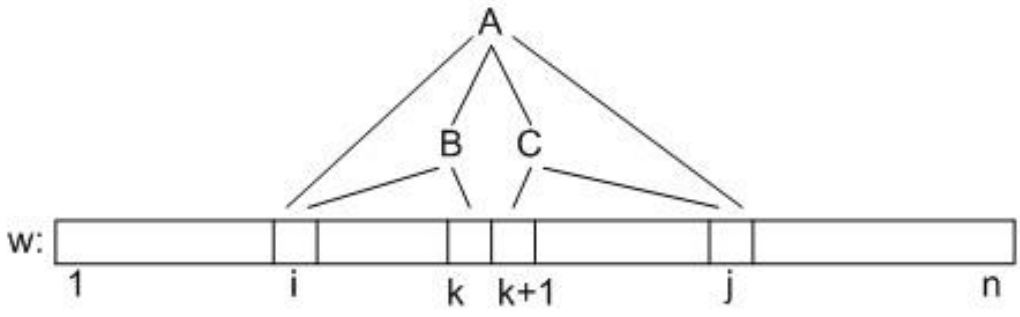

Рис. 1.

Перебор всех подстрок выполняется за $O(n^2)$. Обрабатывая каждую подстроку, алгоритм проводит цикл перебора всех правил грамматики Γ. Такой цикл выполняется за $O(n \cdot |\Gamma|)$. Таким образом при наихудшем исходе время работы алгоритма равно $O(n^3 \cdot |\Gamma|)$ [Hopcroft et al 2001: 300], что делает CYK-алгоритм одним из наиболее эффективных алгоритмов для парсинга наравне с алгоритмом Эрли [Earley 1970], который не требует приведения грамматики к нормальной форме Хомского. Помимо этого, массив d занимает определенный объем памяти.

Способ улучшить эффективность алгоритма путем более эффективного умножения булевых матриц описан в [Valiant 1975]. Применение более эффективных алгоритмов умножения матриц, таких как алгоритм Страссена [Strassen 1969] и алгоритм Копперсмита-Винограда [Coppersmith, Winograd 1990] позволяет далее улучшить время работы алгоритма до $O(n^{2,81} \cdot |\Gamma|)$ и $O(n^{2,376} \cdot |\Gamma|)$ соответственно, однако последний алгоритм показывает хороший результат лишь при работе с матрицами больших размеров [Lee 2002]. В целом же, как показывает [Bodenstab 2009], даже с исходным алгоритмом можно добиться значительных успехов в улучшении эффективности путем оптимизации практической имплементации алгоритма.

2.3.2. Изменения в Helis

Основное изменение в алгоритм CYK, введенное парсером Helis, заключается в замене трехмерного массива фиксированной длины так называемым «линейным индексом», новые элементы в который записываются в процессе парсинга. Линейный индекс для каждой составляющей выдает информацию о том, была ли эта составляющая выведена ранее при выполнении алгоритма, и если так, то какие элементы стояли непосредственно перед ней.

Более того, во избежание комбинаторного взрыва в индекс вписываются не все элементы, а лишь «уникальные». Составляющие считаются уникальными в том случае, если приводят к построению

идентичных структур. Входной текст при парсинге делится на множество пересекающихся подстрок, и такой подход позволяет не сохранять пересекающиеся элементы в памяти несколько раз. Формальным основанием для выбора являются ограничения по типу составляющей и морфологической характеристике, описанные ниже в 2.3.3.

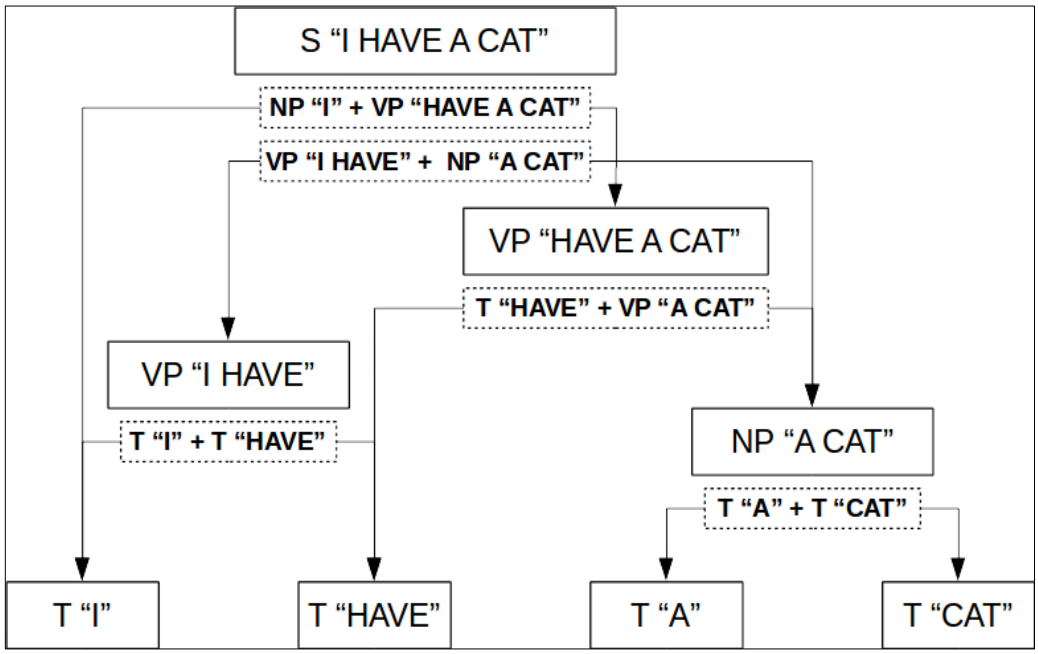

Рис. 2. Усечение пересекающихся элементов

Таким образом Helis сокращает объем памяти, требуемый CYK-алгоритмом для массива *d*, тогда как быстродействие самого алгоритма зависит от степени неоднозначности входных данных.

Стоит отметить, что предлагаемая грамматика является гибридом *грамматики составляющих* и *грамматики зависимостей*. Это строго бинарная грамматика, где вместо обычной последовательности нетерминалов и терминалов указаны два элемента: главный и зависимый. Одна из составляющих маркируется как главный член, тогда как другая составляющая маркируется как зависимый.

2.3.3. Структура синтаксических правил

Рассмотрим структуру правил на конкретном примере:

```
"NP_Adj_Noun": {
   "example": "красивые студенты; большой стол",
```

```
    "head": {
        "type": "NP",
        "tag": "!NoAttributes",
    },
    "subord": {
        "type": "AP",
        "tag": "sg,Case,Number,Gender|pl,Case,Number",
    },
    "target": {
        "type": "NP",
        "copy": { "Head": True },
        "rel": "Attribute",
        "pos": "Left",
    }
}
```

Правила записаны в структуре JSON (JavaScript Object Notation) и состоят из трех основных блоков: *head, subord* и *target*. Они обозначают главный член словосочетания, зависимый член словосочетания и составляющую, образующуюся в результате работы правила, соответственно.

Поле *type* указывает на синтаксический тип составляющей, тогда как в *tag* прописана требуемая от элемента морфологическая характеристика. Теги состоят из последовательностей граммем (пишутся со строчной буквы) и названий словоизменительных категорий (пишутся с заглавной буквы). Последние указывают на то, что данный элемент согласуется с другим в данной категории. Так, в правиле выше главным членом считается группа существительного с любой морфологической характеристикой, тогда как зависимым членом является группа прилагательного, которая *либо* имеет форму единственного числа и согласуется с главным членом в падеже, числе и роде; *либо* имеет форму множественного числа и согласуется с главным членом в падеже и числе.

Вместе они образуют именную группу (“NP” в поле type в блоке target). В поле copy указывается, каким образом формируется морфологическая характеристика новой составляющей. В данном примере она целиком

наследуется от главного члена. В поле *rel* указывается тип связи, а в поле *pos* – позиция зависимого члена. В данном случае правило рассматривает лишь те конструкции, где прилагательное находится в препозиции к существительному.

2.3.4. Шаблоны для извлечения фактов

Helis ограничивается подзадачей извлечения фактов. Факты представляются особой структурой данных типа «ключ-значение», где ключом являются поля факта, и описываются специальными шаблонами. В отличие от простых лексико-синтаксических шаблонов, упомянутых выше в 2.2, шаблоны для извлечения фактов в Helis ищут соответствие с некой частью дерева синтаксического разбора.

Гибридный подход к составлению грамматики, описанный выше, позволяет использовать для описания фактов структуру зависимостей. Поскольку русский язык не имеет строгого порядка элементов (с разной степенью вероятности возможны шесть вариантов: SVO, OSV, VSO, VOS, OVS, SOV), то составление правил, основанных на структуре составляющих, потребовало бы написания отдельного правила для каждого из возможных вариантов. Структура зависимостей, пренебрегающая линейным порядком элементов, позволяет избежать этого.

Нахождение соответствия между синтаксической структурой и шаблоном можно произвести следующим образом: для каждой составляющей генерируется набор всех деревьев зависимостей длины *n*, где *n* – максимальная длина ветви в шаблоне. Каждое полученное таким образом дерево сравнивается с шаблоном. Очевидно, такой подход является затратным, и разработчики предусмотрели ряд оптимизаций, связанных со сверткой генерируемых деревьев при помощи контекстно-независимых правил. В итоге процесс поиска соотношения выполняется простым конечным автоматом. Подробнее о применяемом методе оптимизации можно прочитать в [Popov, Enikeeva 2017].

2.3.5. Структура шаблонов

Рассмотрим структуру шаблонов на конкретном примере:

```
"VerbFrame3": {
   "type": "vf",
   "nodes": [
      { "tag": "Verb", "label": "verb" },
      { "tag": "Noun", "root": 1, "role": "Subj", "label": "subj" },
      { "tag": "Prep", "root": 1, "role": "IObj", "label": "prep" },
      { "tag": "Noun", "root": 3, "role": "Prep", "label": "indobj" },
      { "tag": "Noun", "root": 1, "role": "DObj", "label": "dobj" },
   ],
}
```

Выше представлен шаблон, описывающий структуру трехвалентного глагола. Как и синтаксические правила, шаблоны записаны в структуре JSON. В поле *type* указан тип описываемого факта, а в поле *nodes* – узлы подструктуры зависимостей, с которой и будут сравниваться деревья, генерируемые в основной синтаксической структуре. Для каждого из узлов указан его синтаксический тип и частеречный тег. Атрибут *root* указывает, от какого по счету элемента в шаблоне зависит данный. Так, у первого элемента-глагола этот атрибут отсутствует, потому что глагол является главным членом данной подструктуры. Следующей строкой обозначается деятель-подлежащее, зависящий от глагола. Затем два элемента описывают предложно-падежную группу, которая является косвенным дополнением глагола. Последний же элемент – прямое дополнение.

2.4. Выводы из главы 2

В этой главе мы рассмотрели задачу *извлечения информации* из текста, в которую входят подзадачи *извлечения именованных сущностей* и *извлечения фактов*. Хотя инструменты по извлечению информации для русского языка показывают хороший результат при работе по извлечению

именованных сущностей, извлечение фактов еще не дает надежного результата и остается проблемой, требующей дальнейшего изучения.

Большинство инструментов, работающих с текстами на русском языке, используют подход, основанный на правилах, который отличается высокой точностью и низкой полнотой. Возможной причиной низкой полноты может быть то, что синтаксический парсер отсекает маловероятные верные варианты разбора.

Парсер Helis стремится рассмотреть все варианты разбора и добиться быстродействия благодаря гибридной структуре грамматики и изменениям в алгоритме CYK.

Отдельно были рассмотрены синтаксические правила, в которых указываются главный и зависимый члены, и шаблоны для извлечения фактов, которые накладываются на генерируемую структуру зависимостей.

Глава 3. Составление правил, описывающих синтаксис информационного запроса

3.1. Информационный диалог

Целью этой работы является написание синтаксических правил, которые бы использовались для общения с чат-ботом, выдающим новости по запросу на естественном языке. Для её выполнения следует понимать, каким образом проходит общение между пользователем и чат-ботом и какими отличительными чертами обладает такой вид взаимодействия.

Общение с новостным чат-ботом можно назвать формой **информационного диалога**. А.Е. Кибрик в [Кибрик 1992] определяет информационный диалог как диалог, в котором «объемлющая прагматическая и коммуникативная цель совпадают». То есть прагматическая цель общения сводится к коммуникативной.

3.1.1. Особенности информационного диалога

А.Е. Кибрик изучал записи разговоров людей с оператором справочно-информационной службы 09 и вывел некоторые важные особенности такого типа диалога:

- в них жестко фиксирована прагматико-коммуникативная цель, которая содержится в наличии у пользователя некой **информационной потребности** (а именно, найти телефонный номер определенного абонента);
- пользователь не обучен правилам ведения этого диалога, он лишь приспосабливает к нему свои общие навыки делового общения;
- общение в диалоге ведется на неформализованном разговорном языке;
- коммуникативные роли участников диалога не симметричны.

Были также выделены следующие фазы диалога:

1. Фаза запроса;
2. Фаза нормализации запроса;
3. Фаза ответа.

Во время первой фазы пользователь устанавливает контакт с оператором (обозначает начало диалога, адресата, тип перлокутивной функции речевого акта, намерение вести диалог в соответствии с нормами общения) и вербально выражает свою информационную потребность.

В фазе ответа оператор сообщает искомую пользователем информацию, и последний закрывает диалог посредством клишированных реплик, таких как «*Спасибо*» и «*Благодарю*». Следует учесть, что диалог может окончиться и неудачей.

Вторая фаза – нормализации запроса – является факультативной. Она выполняется лишь в том случае, если пользователь во время первой фазы выразил свой запрос недостаточно ясно, чтобы оператор мог удовлетворить его информационную потребность. В таком случае пользователю задаются промежуточные уточняющие вопросы.

Формат информационного диалога хорошо описывает интеракцию между пользователем и чат-ботом, хотя, очевидно, следует учесть некоторые различия. Во-первых, общение с чат-ботом осуществляется в письменном виде, хотя и отличается неформальным, разговорным характером. Во-вторых, разговор лишен структурного элемента. Хотя в общих чертах деление на три фазы справедливо и для диалога с чат-ботом, пользователь определенно не будет обозначать начало диалога отдельным лексическим элементом, не будет эксплицитно обращаться к адресату и вряд ли будет использовать типичные для делового общения формы вежливости («*Скажите, пожалуйста*»). Равно как и верно то, что пользователь не станет формально закрывать диалог. Пользователи, как правило, ограничиваются функциональной стороной диалога. Они лишь требуют ответа на запрос и не ожидают диалога, похожего на общение с человеком, от системы, созданной для выполнения конкретной задачи [Kurilchik 2017].

С другой стороны, фаза нормализации запроса остается релевантной. Чат-бот, выполняющий конкретную функцию, требует определенного набора

аргументов, и может задать пользователю уточняющий вопрос, если тот не указал некоторые из них.

3.1.2. Формальный анализ запроса пользователя

А.Е. Кибрик не стремился создать работающую диалоговую систему на основе своих наблюдений. Пример подобного исследования, но ориентированного на практическое применение можно обнаружить в [Страндсон 2008]. Данное исследование основано на Эстонском диалоговом корпусе, а именно на записях около 1000 звонков в справочную службу. Авторы используют подход, схожий с теорией Кибрика, и глубже описывают диалог с точки зрения конверсационного анализа. Так, утверждается, что диалоговые акты могут образовывать смежные пары, где первый элемент обуславливает появление второго: информационный запрос обуславливает появление ответа на него, и компьютеру важно понять рамки запроса в текущем диалоге.

Сам запрос может иметь форму **вопроса** или **директива**. В [Koit et al. 2006] проводится различие между двумя категориями: в вопросах используются такие формальные черты как вопросительные слова и особый порядок слов. В [Страндсон 2008] описан формат диалога, имеющего структуру «директив – выполнение директива», где директив выражает желание или потребность пользователя получить некую информацию («*Подскажите мне последний поезд на Тарту*»).

Исследования авторов показали, что наиболее часто директивы выражаются глаголами ütlema (‘сказать’) в форме императива и soovima (‘желать’) в форме кондиционалиса (сослагательного наклонения). Полное распределение форм пяти наиболее частотных глаголов выглядит так:

ГЛАГОЛ	НАКЛОНЕНИЕ (#)			ВСЕГО
	инд.	конд.	имп.	
paluma ’просить’ (в 1 л. ед.ч.	6	8		14

«пожалуйста»)				
ütlema ’сказать’			8	8
soovima ’желать' («я хочу/ хотел бы»)	2	5		7
tahtma ’хотеть’		5		5
võtma ’брать’(«дайте»)		5		5

Таблица 2.

К сожалению, столь детальные исследования на материале русского языка еще не проводились. Поэтому в рамках этой работы была собрана небольшая выборка потенциальных запросов новостному чат-боту на русском языке (Прил. 1). Респондентам был задан следующий вопрос: «*Допустим, что Вы говорите с чат-ботом, который может выдать новость с определенного сайта из следующего перечня (Эхо Москвы, Фонтанка.ру, Интерфакс, Известия, Лента.ру, Медуза, newsru, РБК, ТАСС, Ведомости) по свободным темам. Вы можете не указывать конкретный сайт. Ваш запрос является не фиксированной командой, а предложением на естественном языке. Приведите пример подобного запроса*».

Было собрано 80 ответов, которые можно разделить на 3 группы:

1. Директивы («Хочу новости про чемпионат по футболу», «покажи экономические новости с фонтанки ру», «Дай новость о переговорах Меркель с Путиным») [31 вхождение];
2. Вопросы («Кто победил на выборах президента 2018», «Почему закрыт Невский?», «в каких городах проходит чм по футболу 2018») [35 вхождений];
3. Группы ключевых слов («Отравление Скрипаля», «Открытие крымского моста», «последние политические новости») [14 вхождений].

Полный список ответов приведен в Приложении 1.

3.2. Описание правил

3.2.1. Директивы

Рассмотрим директивы как базовую модель запроса пользователя. На основе данных можно составить таблицу, аналогичную Таблице 2:

ГЛАГОЛ	НАКЛОНЕНИЕ (#)			ВСЕГО
	инд.	конд.	имп.	
показать			9	9
дать			6	6
хотеть	5			5
сказать			4	4
рассказать			3	3

Таблица 3. Распределение пяти наиболее частотных глагольных форм среди запросов на русском языке

Можно заметить, что в русском языке преобладают формы императива, и, хотя в выборке не было форм сослагательного наклонения, ими также возможно выразить директив («мне бы хотелось новость», «я хотел бы»).

Примечательно, что объектом глагола, вводящего директив, могут быть как слова, обозначающие новостной материал («новость», «статья»), так и сами сайты источники («Покажи Известия»). Есть случай управления инфинитивом («Хочу почитать Ведомости») и предложно-падежной группой («расскажи про спорт»).

Опишем случай присоединения дополнения к переходным глагольным формам простым правилом (все указанные правила приведены в Прил. 2):

```
"VerbObject": {
    "example": "покажи Известия, хочу почитать, почитать Ведомости, расскажи про спорт",
    "head": {
        "tag": "VP, tran|INFN, tran"
    },
    "subord": {
        "tag": "NP, accs|ADFS, accs|NPRO, accs|INFN|PP",
    },
    "target": {
        "copy": { "Head": True },
```

```
            "pos": "Right",
        }
    },
```

Поскольку переходные глаголы в морфологическом словаре имеют особую помету “tran”, в правиле можно указать, что элемент, указанный как главный, должен иметь эту помету, тогда как зависимый элемент должен находиться в форме винительного падежа (помета “accs”) или же являться инфинитивом или предложной группой (помета класса “PP”).

Схожим образом можно ограничить подчинительную связь управления существительных. Хотя эта связь имеет множество способов выражения (статистически доминирующий родительный падеж, а также дательный и творительный; притом, что наблюдается вариативность всех трех форм с присоединением предложно-падежных групп), в рассматриваемой ситуации можно ограничиться случаем присоединения существительного в родительном падеже. Например: «Покажи *статьи РБК* про конфликт в Сирии»:

```
"NounObject": {
        "example": "статьи РБК",
        "head": {
            "tag": "NP"
        },
        "subord": {
            "tag": "NP, gent",
        },
        "target": {
            "copy": { "Head": True },
            "pos": "Right",
        }
    },
```

Сложнее описать присоединение предложно-падежных групп. Поскольку правила пишутся с опорой на грамматику зависимостей, то триплет «существительное-предлог-существительное» («новость про конфликт») приходится разделить на пары, описываемые разными

правилами. Одно правило просто допускает возможность присоединения к существительному предложной группы справа. Другое же описывает связь предлога с зависимым существительным. Так как в данном случае нельзя предсказать конкретные морфологические характеристики зависимого элемента, можно лишь обозначить, что он стоит в косвенном падеже. Итоговое правило:

```
"PrepPhrase": {
        "example": "про президента, об экономике, с сайта",
        "head": {
                "tag": "PREP"
        },
        "subord": {
                "tag": "NP,!nomn",
        },
        "target": {
                "tag": "PP",
                "copy": { "Head": True },
                "pos": "Right",
        }
},
```

В таблице 3 указано, что глагол «хотеть» используется в формах индикатива. Подлежащее «Я» в таких случаях зачастую опускается, но может оставаться на месте. Требуется общее правило для формирования ядра предложения: подлежащего и сказуемого (в том числе и именного):

```
"Sentence": {
        "example": "я хочу, кто выиграл",
        "head": {
                "tag": "VP|NP, nomn|ADJS|PRTS,pssv"
        },
        "subord": {
                "tag": "NP, nomn, NMbr|NPRO, nomn, NMbr",
        },
        "target": {
                "tag": "S",
                "copy": { "Head": True },
```

```
                "pos": "Both",
            }
        },
```

В правиле указано, что подлежащим может быть только существительное или местоимение-существительное в именительном падеже, согласующееся со сказуемым в числе. Сказуемое же может быть выражено глагольной группой, а также существительным в именительном падеже, кратким прилагательным или пассивным кратким причастием.

Однако среди ответов также встретилась конструкция именного сказуемого без грамматически выраженного подлежащего: «Мне нужны новости про чемпионат по футболу». Этот случай требует написания отдельных правил:

```
"AgentAdjNeed": {
        "head": {
            "tag": "ADJS,Qual"
        },
        "subord": {
            "tag": "NPRO,datv",
        },
        "target": {
            "copy": { "Head": True },
            "pos": "Left",
        }
},
```

```
"AdjNeedObject": {
        "head": {
            "tag": "ADJS,Qual"
        },
        "subord": {
            "tag": "NP, NMbr, accs",
        },
        "target": {
            "copy": { "Head": True },
            "pos": "Right",
        }
},
```

Первое правило описывает словосочетание «мне нужен»: присоединение личного местоимения в дательном падеже к краткому качественному прилагательному. Второе правило описывает присоединение к этому прилагательному дополнения, выраженного существительным в винительном падеже, согласующимся в числе с главным элементом.

Возможно и присоединение придаточных предложений. Например: «Скажи, чем занят Медведев» или «Покажи новость, где есть Путин». Присоединение придаточных также происходит в два этапа: присоединение придаточного предложения к союзу или к союзному слову, в результате чего

полученная составляющая маркируется пометой “S”, и присоединение этой группы к концу главного предложения. Стоит отметить, что, если союзное слово выступает также и подлежащим придаточного («Скажи, кто выиграл на Евровидении»), правило “EnterSub” допускает прямое присоединение придаточного.

```
"ConjSub": {
        "head": {
                "tag": "CONJ|ADVB,Ques"
        },
        "subord": {
                "tag": "S",
        },
        "target": {
                "tag": "S",
                "copy": { "Head": True },
                "pos": "Right",
        }
},
```

```
"EnterSub": {
        "head": {
                "tag": "VP|NP"
        },
        "subord": {
                "tag": "CONJ,S|ADVB,Ques,S|S",
        },
        "target": {
                "copy": { "Head": True },
                "pos": "Right",
        }
},
```

Встретилось также и использование закрытого ряда с отрицательно-противительным значением для уточнения запроса, указания взаимного исключения тем в искомой новости («покажи новости про москву но не про собянина»). Такая конструкция может быть также выражена союзами «не… а» («а… не») и «не… но» [Шведова 1980: 170]. Постараемся описать такие случаи следующими правилами:

```
"Adversative": {
        "example": "но не собянина", "а про медведева",
        "head": {
                "tag": "CONJ"
        },
        "subord": {
                "tag": "NP, !S|PP, !S|AP, !S",
        },
        "target": {
                "copy": { "Head": True },
```

```
"Adversative_Enter": {
        "example": "про москву но", "не про путина а",
        "head": {
                "tag": "NP|PP|AP"
        },
        "subord": {
                "tag": "CONJ, !S",
        },
        "target": {
                "copy": { "Head": True },
```

```
            "pos": "Right",
        }
},
```

```
            "pos": "Right",
        }
},
```

К сожалению, морфологический словарь не приводит детального деления союзов на разряды, поэтому мы можем использовать лишь общие правила для присоединения союза к именным или предложно-падежным группам, что описывает закрытые союзные ряды вообще («“Новокрестовская” и “Беговая”»). Чтобы отличить этот набор правил от правил для присоединения придаточных, приведенных ниже, можно обозначить, что присоединяемый оборот и вводящий его союз не обладают маркером “S”.

Также требуется описать связь согласования между прилагательным и существительным («популярные новости») и случай присоединения наречия степени к качественному прилагательному («*наиболее* популярные новости»):

```
"ConcordAdjNoun": {
        "example": "последние новости",
        "head": {
            "tag": "NP,sing,GNdr|NP,plur"
        },
        "subord": {
            "tag": "AP,CAse,NMbr",
        },
        "target": {
            "copy": { "Head": True },
            "pos": "Left",
        }
},
```

```
"AdverbialMod": {
        "example": "наиболее популярные",
        "head": {
            "tag": "AP,Qual"
        },
        "subord": {
            "tag": "AdvP, !Ques",
        },
        "target": {
            "copy": { "Head": True },
            "pos": "Left",
        }
},
```

Первое правило описывает согласование прилагательного с существительным в падеже, числе и роде (если существительное имеет форму единственного числа). Второе правило просто описывает присоединение невопросительного наречия слева к качественному прилагательному.

Так как в запросе могут быть указаны конкретные личности, требуется правило, которое бы описывало согласование имени и фамилии. Поскольку в морфологическом словаре есть отдельные пометы “Name” и “Surn”, можно легко указать имя главным членом словосочетания, а фамилию – зависимым, который согласуется с главным по падежу, роду и числу:

```
"FullName": {
        "example": "Владимир Путин, Ангела Меркель",
        "head": {
                "tag": "NP, Name"
        },
        "subord": {
                "tag": "NP, Surn, GNdr, NMbr, CAse",
        },
        "target": {
                "copy": { "Head": True },
                "pos": "Right",
        }
},
```

Следует также отметить, что в запросах пользователя могут, хоть и редко, встречаться формы вежливости («покажи, пожалуйста, наиболее популярные новости за последний час»). Хотя место вводных слов в предложении свободное [Шведова 1980: 230], нас прежде всего интересуют случаи, когда оно непосредственно предшествует глаголу или следует за ним. Глагол при этом всегда стоит в форме императива.

```
"Please": {
        "example": "пожалуйста покажи",
        "head": {
                "tag": "VP, impr|NP|PP"
        },
        "subord": {
                "tag": "INTJ",
        },
        "target": {
                "copy": { "Head": True },
                "pos": "Both",
```

```
        }
},
```

Данное правило также допускает присоединение вводного слова к именной или предложно-падежной группе (например, в самом конце запроса, после обозначения темы). Правило использует морфологическую помету “INTJ” (междометие) для определения зависимого элемента, что допускает возможность применения правила в неподходящих ситуациях (когда вместо «пожалуйста» встречается другой элемент с пометой “INTJ”). Однако вероятность употребления других междометий в запросах довольно мала. Так, в собранных ответах не было обнаружено подобных случаев.

Также среди запросов нет случаев обращения, хотя стоит учесть, что если система будет названа определенным именем, то пользователь вполне может использовать его в своих репликах. Обращение либо открывает, либо закрывает предложение [Шведова 1980: 164]. На этом основывается следующее правило:

```
"Address": {
        "example": "яндекс, покажи",
        "head": {
                "tag": "VP, impr|NP|PP"
        },
        "subord": {
                "tag": "NP, nomn, Name|NP, nomn, Orgn",
        },
        "target": {
                "copy": { "Head": True },
                "pos": "Both",
        }
},
```

Обращение может либо непосредственно предшествовать глаголу в форме императива [Шведова 1980: 163] и находиться в начале предложения, либо закрывать предложение после некой именной группы. Само обращение выражено существительным в именительном падеже, причем в

морфологическом словаре оно должно сопровождаться пометой “Name” (для имени) или “Orgn” (для названия организации, продукта, сервиса).

Итак, для директивов были созданы правила, описывающие следующие конструкции: дополнение глагола, дополнение существительного, присоединение предлога к существительному, подлежащее и сказуемое, конструкции типа «мне нужны новости», присоединение придаточных предложений, присоединение отрицательно-противительного оборота, согласование прилагательных с существительными, присоединение наречий к качественным прилагательным, согласование имени и фамилии, использование вводного слова «пожалуйста» как формы вежливости и обращение.

3.2.2. Вопросы

Выше было указано, что вопросы отличаются от директивов использованием вопросительных частиц и особого порядка слов. Вопросительные предложения в русском языке зачастую являются трансформами невопросительных [Шведова 1980: 387] с измененной интонацией или присоединенной вопросительной частицей. Порядок слов в вопросительных предложениях в русском языке отличается большей свободой, чем в невопросительных, особенно если вопрос задается интонационно («Саша *хорошо* учится?», «Саша учится *хорошо*?», «*Хорошо* Саша учится?», «*Хорошо* учится Саша?») [Шведова 1980: 396] Для письменной же речи, где вопрос задается при помощи введения вопросительных частиц, порядок речи обычно является обратным стилистически нейтральному, где вопросительное слово стоит в начале предложения непосредственно перед глаголом («Куда уехали дети? — Дети уехали в деревню»). На основе этих закономерностей было написано следующее правило, описывающее последовательность «вопросительное слово-сказуемое»:

```
"QuestionVerbObject": {
```

```
        "example": "чем занят, когда откроют, на сколько закрыт, куда пойти",
        "head": {
                "tag": "VP|INFN|ADJS|PRTS,pssv"
        },
        "subord": {
                "tag": "NPRO|ADVB,Ques|PP",
        },
        "target": {
                "copy": { "Head": True },
                "pos": "Left",
        }
},
```

Сказуемое в вопросе может быть выражено глагольной группой, но есть и случаи использования именного сказуемого, которое может быть выражено инфинитивом, краткими прилагательным и причастием. Вопросительное слово же может быть выражено местоимением, вопросительным наречием, либо же предложной группой.

3.2.3. Группы ключевых слов

Такие запросы, как «Отравление Скрипаля», «Новости РБК» и «последние новости петербурга», больше похожи на команды, указывающие конкретные темы, интересующие пользователя, чем на естественные запросы. Однако они, по большей части, все равно выражены на естественном языке и поддаются синтаксическому анализу. Так как их структура довольно проста (это назывные предложения, состоящие из именных групп), их можно описать правилами, уже приведенными выше в 3.2.1., а именно “NounObject” и “ConcordAdjNoun”. Первое правило описывает управление существительного («Отравление Скрипаля», «Новости РБК», «новости петербурга»), а второе – согласование прилагательного с существительным («последние новости»).

3.3. Проверка результатов и недостатки

Описанные выше правила должны справиться с выделением синтаксической структуры в обозначенных запросах пользователя. Однако они безусловно не покрывают все возможные вариации, которыми человек может выразить свою информационную потребность. В запросах на многих позициях может встретиться эллипсис («когда [проходит] юридический форум», «новость про про Москву, но не [про] Собянина»), и предугадать все такие случаи является крайне сложной задачей, тогда как написание правил для каждого из них увеличит количество вариантов разбора, что излишне усложнит работу системы.

Все 80 запросов из Прил. 1 были обработаны парсером Helis, и 26 предложений (~33% выборки) по тем или иным причинам не были обработаны полностью, т.е. среди вариантов разбора не было единого дерева, включавшего в себя все токены предложения. Почти половина ошибок (а именно 12) возникла из-за отсутствия определенных словоформ, таких как «Скрипаль», «Мутко», «КАД», «Оно» (как фамилия), в морфологическом словаре. Затем наибольшее число неверных разборов пришлось на случаи эллипсиса в вопросах («Что интересного на Ленте?», «что там нового про футбол»). Несколько раз парсер производил некорректную токенизацию слов, в состав которых входит дефис («Санкт-Петербург», «что-нибудь»). В запросах «какие у тебя есть новости из интерфакса?» и «Какие есть новости в области медицины?» вопросительное слово «какие», вынесенное в начало предложения, не удалось соотнести с токеном «новости», так как их разделяет сказуемое «есть».

Выявленные ошибки в целом относятся к обработке вопросительных предложений и поднимают отдельную проблему формального описания синтаксических трансформов. С другой стороны, директивы, установленные выше как базовая форма запроса, были обработаны в полном объеме.

Однако даже в случаях ошибок Helis позволяет избежать полной остановки работы, так как он выдает и неполные варианты разбора. Даже если некий элемент не позволит собрать полное дерево зависимостей (как в

случае запроса «Что нового в Западной Европе?», где токен «что» оказывается отсоединен от остальной структуры), среди вариантов разбора окажутся отдельные поддеревья, на которые все еще будет возможно наложить шаблон.

3.4. Выводы из главы 3

В этой главе было рассмотрено понятие *информационного диалога*, а также был дан краткий обзор корпусных исследований его формальной и синтаксической структур.

В информационном диалоге пользователь выражает свою *информационную потребность* двумя способами: вопросом или директивом, где первый отличается использованием вопросительных частиц и особым порядком слов.

Были собраны образцы возможных запросов пользователей новостному чат-боту. Большинство примеров относится либо к вопросам, либо директивам, причем эти два типа имеют примерно одинаковую частоту. На основе собранных данных и имеющихся академических исследований синтаксиса русского языка были созданы синтаксические правила для парсера Helis, которые ориентированы на пользовательский запрос новостному чат-боту. Наиболее существенным из них было дано описание и обоснование.

Хотя данные правила не могут рассчитывать на стопроцентную точность анализа (по причине многообразия способов выражения информационной потребности в русском языке), Helis все равно может применить их к отдельным подстрокам запроса и получить неполный анализ, с которым все еще можно продолжить работу.

Заключение

В этой работе были представлены правила синтаксического анализа в грамматике парсера Helis для обработки запросов пользователя на естественном языке, обращенных новостному чат-боту.

Было написано 20 правил, вошедших в итоговый набор, 16 из которых были описаны в главе 3. Большинство правил было составлено по модели запроса-директива, так как этот тип предложений показывает наибольшую вариативность основных синтаксических структур, тогда как другой частотный тип запросов – вопрос – можно описать трансформацией имеющихся структур. Наименее частотный третий тип запросов из выборки в свою очередь полностью покрывается правилами, описывающими подчинительные связи существительных.

В процессе работы было отмечено, что наиболее часто к большому количеству вариантов разбора приводит морфологическая неоднозначность. В используемом парсером словаре OpenCorpora встречаются целые парадигмы неизменяемых словоформ, которые могут совпадать с другими лексемами (например, предлог «про» и аббревиатура «ПРО», личное местоимение «я» и фамилия «Я»). Хотя это явление не привело к случаям комбинаторного взрыва при работе, работа парсера на синтаксическом уровне может быть улучшена на предваряющем его этапе морфоанализа.

Итоговые правила не стремятся исчерпывающе описать все вариации запросов на естественном языке. Без объёмного корпуса логов общения с подобными системами такая задача является чрезвычайно сложной для выполнения. Запрос может быть сформулирован так, что парсер не окажется способен составить единое древо разбора. Так, при проверке результатов было выявлено, что большинство ошибок при анализе возникает из-за того, что лексический модуль парсера не распознает определенные токены. Помимо этого, проблемы создал синтаксис вопросительных предложений, в которых изменяется порядок слов и частотен эллипсис. Однако директивы, базовый тип запросов, не выявили ошибок в работе парсера. И поскольку

Helis выдает все варианты анализа, в том числе и неполные, результаты всё равно передаются на следующий уровень, где к полученным поддеревьям будут применяться шаблоны для извлечения фактов.

Литература

1. *Bird S. et al.* Natural Language Processing with Python. O'Reilly, 2009. Ch. 7
2. *Bodenstab N.* Efficient Implementation of the CKY algorithm. 2009. URL: https://pdfs.semanticscholar.org/b2e3/0053b54df93e39edbdd84d4e5611252110c5.pdf
3. *Cer D. et al.* Universal Sentence Encoder // arXiv:1803.11175, 2018. URL: https://arxiv.org/abs/1803.11175
4. *Colby et al.* A Computer Method of Psychotherapy // The Journal of Nervous and Mental Disease, 142(2), 1966. pp. 148-152
5. *Coppersmith D., Winograd S.* Matrix multiplication via arithmetic progressions // Journal of Symbolic Computation, 9(3), 1990. pp. 251-280
6. *Earley J.* An efficient context-free parsing algorithm // Communications of the ACM, 13(2), 1970. pp. 94-102
7. *Hearst, M. A.* Automatic acquisition of hyponyms from large text corpora // Proceedings of COLING-92. Association for Computational Linguistics, 1992. pp. 539-545
8. *Hill J. et al.* Real conversations with artificial intelligence: A comparison between human–human online conversations and human–chatbot conversations // Computers in Human Behavior, 49, 2015. pp. 245-250
9. *Hopcroft J. et al.* Introduction to Automata Theory, Languages, and Computation, Second Edition. Addison-Wesley, 2001
10. *Jurafski D., Martin J.* Speech and Language Processing, Third edition. 2017, Ch. 12, 21, 28, 29. URL: https://web.stanford.edu/~jurafsky/slp3/
11. *Kasami T.* An efficient recognition and syntax analysis algorithm for context-free languages. Air Force Cambridge Research Laboratory, 1965
12. *Koit M. et al.* Processing Customer Requests: An Analysis of the Estonian Dialogue Corpus // Proceedings of the 11th International Conference "Speech and Computer" SPECOM'2006. Anatolya Publishers, 2006. pp. 193-198

13. *Kurilchik E.* Chatbots as a Digital Marketing Communication Tool. 2017. URL: http://www.theseus.fi/handle/10024/131171
14. *Lally A., Fodor P.* Natural language processing with Prolog in the IBM Watson system // The Association for Logic Programming (ALP) Newsletter, 2011
15. *Lee L.* Fast context-free grammar parsing requires fast boolean matrix multiplication // Journal of the ACM, 49(1), 2002. pp. 1-15
16. *Pinker S.* The Language Instinct. HarperPerennial, 1995. pp. 192-228
17. *Popov A., Enikeeva E.* Template Search Algorithm for Multiple Syntactic Parses // Proceedings of IMS-2017. ACM Press, 2017. pp. 164-171
18. *Schank R., Abelson R.* Scripts, plans, goals and understanding: An inquiry into human knowledge structures. Lawrence Erlbaum, 1977
19. *Sipser M.* Introduction to the Theory of Computation. Thomson Course Technology, 1997. pp. 99-135
20. *Strassen V.* Gaussian elimination is not optimal // Numerische Mathematik, 14(3), 1969. pp. 354-356
21. *Turing A.* Computing Machinery and Intelligence // Mind, LIX(236), 1950. pp. 433-460
22. *Valiant L.* General context-free recognition in less than cubic time // Journal of Computer and System Sciences, 10(2), 1975. pp. 308-314
23. *Warwick K., Shah H.* Can machines think? A report on Turing test experiments at the Royal Society // Journal of experimental & Theoretical artificial Intelligence, 28(6), 2016. pp. 989-1007
24. *Weizenbaum, J.* ELIZA – A computer program for the study of natural language communication between man and machine // Communications of the ACM, 9(1), 1966. pp. 36-45
25. *Younger D.* Recognition and parsing of context-free languages in time n^3// Information and Control, 10(2), 1967. pp. 189-208
26. *Добров А.В.* Компьютерный синтаксис // Прикладная и компьютерная лингвистика. М.: URSS, 2016

27. *Кибрик А.Е.* Очерки по общим и прикладным вопросам языкознания (универсальное, типовое и специфичное в языке). М.: МГУ, 1992. С. 301-313
28. *Константинов Н.С., Дегтева А.В.* Диалоги и чат-боты // Прикладная и компьютерная лингвистика. М.: URSS, 2016
29. *Старостин А.С. и др.* FactRuEval 2016: Тестирование систем выделения именованных сущностей и фактов для русского языка // Труды ежегодной Международной конференции «Диалог 2016». М.: РГГУ, 2016. С. 702-721
30. *Страндсон К. и др.* К взаимодействию компьютера и человека на естественном языке // Труды ежегодной Международной конференции «Диалог». М.: РГГУ, 2008. С. 495-503
31. *Шведова Н.Ю. и др.* Русская грамматика. Том II. Синтаксис. М.: Наука, 1980

Электронные ресурсы

1. Automatic Content Extraction // ACE | Linguistic Data Consortium. URL: https://www.ldc.upenn.edu/collaborations/past-projects/ace (последнее обращение 23.05.2018)
2. Cleverbot // Wikipedia, the free encyclopedia. URL: https://en.wikipedia.org/wiki/Cleverbot (последнее обращение 23.05.2018)
3. CYK algorithm // Wikipedia, the free encyclopedia. URL: https://en.wikipedia.org/wiki/CYK_algorithm (последнее обращение 23.05.2018)
4. ELIZA // Wikipedia, the free encyclopedia. URL: https://en.wikipedia.org/wiki/ELIZA (последнее обращение 23.05.2018)
5. FactRuEval 2016. // Конференция Диалог 2016. URL: http://www.dialog-21.ru/evaluation/2016/ner/ (последнее обращение 23.05.2018)
6. Helis // Hedge Linguistic System. URL: http://hurmining.com/helis (последнее обращение 23.05.2018)
7. Talk to Books // Google Книги. URL: https://books.google.com/talktobooks/ (последнее обращение 23.05.2018)
8. Turing Test // Wikipedia, the free encyclopedia. URL: https://en.wikipedia.org/wiki/Turing_test (последнее обращение 23.05.2018)
9. Watson (computer) // Wikipedia, the free encyclopedia. URL: https://en.wikipedia.org/wiki/Watson_(computer) (последнее обращение 23.05.2018)
10. Алгоритм Кока-Янгера-Касами разбора грамматики в НФХ // Сайт вики-конспектов Университета ИТМО. URL: http://neerc.ifmo.ru/wiki/index.php?title=Алгоритм_Кока-Янгера-Касами_разбора_грамматики_в_НФХ (последнее обращение 23.05.2018)

11. Инструкция по определению именованных сущностей // OpenCorpora Вики. URL: http://opencorpora.org/wiki/Nermanual/1 (последнее обращение 23.05.2018)
12. Томита-парсер // Технологии Яндекса. URL: https://tech.yandex.ru/tomita/ (последнее обращение 23.05.2018)

Приложение 1. Собранные потенциальные запросы пользователей новостному чат-боту

1. Директивы:

Покажи новость про Северную Корею.

Хочу новости про чемпионат

Хочу новости про чемпионат по футболу

Дай новость с Эха

Покажи Известия

Покажи Ленту

Хочу почитать Ведомости

Найди информацию о последних санкциях против России

Дай новость о переговорах Меркель с Путиным

Мне нужны новости про чемпионат по футболу

Скажи, кто выиграл на Евровидении

Скажи, чем занят Медведев

Скажи, когда юридический форум

Скажи, куда пойти на выходные

расскажи про спорт

расскажи про фифу

покажи новости про москву но не про собянина

покажи экономические новости с фонтанки ру

покажи, пожалуйста, наиболее популярные новости за последний час

Расскажите о событиях в Сирии

Подскажите имя нового министра здравоохранения

Покажи новость о сегодняшнем курсе доллара

Дай новость о Каннском фестивале

Выдай новость о результатах чемпионата мира по хоккею

Дай статью РБК про санкции

Покажи что-нибудь про Украину.

покажи новость про Мутко

хочу новости экономики

Дай последнее с Ведомостей.

Хочу новости про Скрипаля.

дай известия про выборы

2. Вопросы:

Кто выиграл Евровидение?

Что нового на Медузе?

Что интересного на Ленте?

Когда будет ЧМ по футболу?

Сколько лет Шер?

Сколько стоит проезд по подорожнику

Во сколько темнеет?

Как здоровье у Скрипаля?

Кого из министров уволили?

Кто победил на выборах президента 2018

Почему закрыт Невский?

Когда откроют Невский?

в каких городах проходит чм по футболу 2018

Как там Скрипали поживают?

Что произошло на КАД?

Россия или Канада?

Когда откроют «Новокрестовскую» и «Беговую» для простых смертных?

Кого озвучивала Йоко Оно в «Острове собак»?

что там нового про футбол

чего нового в мире политики?

что произошло за ночь в спб

что произошло за ночь в спб

что творится в этом мире

какие у тебя есть новости из интерфакса?

что нового пишет медуза?

Что произошло сегодня?

Какие события произошли за последний час?

Есть ли новости про правительство РФ?

Что случилось сегодня в Санкт-Петербурге?

Что произошло в Москве?

Какие есть новости в области медицины?

Что нового в Западной Европе?

Запретили ли что-нибудь в последнее время?

Что нового на Украине?

Какие законы приняты сегодня?

3. Группы ключевых слов:

Отравление Скрипаля

Блокировка телеграм

Навальный

Митинг Навального

Открытие крымского моста

Новости Лентач

Новости РБК

Крушение самолета на Кубе

Он нам не царь 2019

матч германия испания какой счет

последние политические новости

последние новости петербурга

политические новости медузы за прошедший день

последние спортивные события

Приложение 2. Правила синтаксического анализа запросов пользователя

1. Подлежащее и сказуемое

```
"Sentence": {
        "example": "я хочу, кто выиграл",
        "head": {
                "tag": "VP|NP, nomn|ADJS|PRTS,pssv"
        },
        "subord": {
                "tag": "NP, nomn, NMbr|NPRO, nomn, NMbr",
        },
        "target": {
                "tag": "S",
                "copy": { "Head": True },
                "pos": "Both",
        }
},
```

2. Дополнение переходного глагола

```
"VerbObject": {
        "example": " покажи Известия, хочу почитать, почитать Ведомости, расскажи про спорт ",
        "head": {
                "tag": "VP, tran|INFN, tran"
        },
        "subord": {
                "tag": "NP, accs|INFN|PP|NPRO, accs|ADFS, accs",
```

```
        },
        "target": {
            "copy": { "Head": True },
            "pos": "Both",
        }
    },
```

3. Дополнение непереходного глагола

```
    "VerbIndirObject": {
        "example": "произошло в Москве",
        "head": {
            "tag": "VP, intr|INFN, intr"
        },
        "subord": {
            "tag": "PP|INFN",
        },
        "target": {
            "copy": { "Head": True },
            "pos": "Both",
        }
    },
```

4. Последовательность «вопросительное слово + сказуемое», открывающая вопрос

```
    "QuestionVerbObject": {
        "example": "чем занят, когда откроют, на сколько закрыт, куда пойти",
        "head": {
            "tag": "VP|INFN|ADJS|PRTS,pssv"
```

```
        },
        "subord": {
            "tag": "NPRO|ADVB,Ques|PP",
        },
        "target": {
            "copy": { "Head": True },
            "pos": "Left",
        }
    },
```

5. Дополнение существительного

```
    "NounObject": {
        "example": "статьи РБК",
        "head": {
            "tag": "NP"
        },
        "subord": {
            "tag": "NP, gent",
        },
        "target": {
            "copy": { "Head": True },
            "pos": "Right",
        }
    },
```

6. Присоединение частицы

```
    "Particle": {
        "head": {
```

```
            "tag": "NP|PP|VP"
        },
        "subord": {
            "tag": "PRCL",
        },
        "target": {
            "copy": { "Head": True },
            "pos": "Both",
        }
    },
```

7. Присоединение предлога

```
    "PrepPhrase": {
        "example": "про президента, об экономике, с сайта",
        "head": {
            "tag": "PREP,!PP"
        },
        "subord": {
            "tag": "NP,!nomn|AP,!nomn|NPRO,!nomn|NUMR",
        },
        "target": {
            "tag": "PP",
            "copy": { "Head": True },
            "pos": "Right",
        }
    },
```

8. Косвенное дополнение существительного

```
"PrepObject": {
        "head": {
                "tag": "NP|NPRO"
        },
        "subord": {
                "tag": "PP",
        },
        "target": {
                "copy": { "Head": True },
                "pos": "Right",
        }
},
```

9. Присоединение наречия

```
"AdverbMod": {
        "example": "наиболее популярные",
        "head": {
                "tag": "VP|AP,Qual|PRTS,pssv"
        },
        "subord": {
                "tag": "AdvP, !Ques",
        },
        "target": {
                "copy": { "Head": True },
                "pos": "Left",
        }
},
```

10. Согласование прилагательного с существительным

```
"ConcordAdjNoun": {
        "example": "последние новости",
        "head": {
                "tag": "NP,sing,GNdr|NP,plur"
        },
        "subord": {
                "tag": "AP,CAse,NMbr",
        },
        "target": {
                "copy": { "Head": True },
                "pos": "Left",
        }
},
```

11. Присоединение придаточного предложения к главному

```
"EnterSub": {
        "head": {
                "tag": "VP|NP"
        },
        "subord": {
                "tag": "CONJ,S|ADVB,Ques,S|S",
        },
        "target": {
                "copy": { "Head": True },
                "pos": "Right",
        }
},
```

12. Образование составляющей, включающей в себя собственно придаточное и союз/союзное слово

```
"ConjSub": {
        "head": {
                "tag": "CONJ|ADVB,Ques"
        },
        "subord": {
                "tag": "S",
        },
        "target": {
                "tag": "S",
                "copy": { "Head": True },
                "pos": "Right",
        }
},
```

13. Отрицательно-противительный оборот

```
"Adversative": {
        "example": "но не собянина", "а про медведева",
        "head": {
                "tag": "CONJ"
        },
        "subord": {
                "tag": "NP, !S|PP, !S|AP, !S",
        },
        "target": {
                "copy": { "Head": True },
```

```
            "pos": "Right",
        }
    },
```

14. Присоединение отрицательно-противительного оборота

```
    "Adversative_Enter": {
        "example": "про москву но", "не про путина а",
        "head": {
            "tag": "NP|PP|AP"
        },
        "subord": {
            "tag": "CONJ, !S",
        },
        "target": {
            "copy": { "Head": True },
            "pos": "Right",
        }
    },
```

15. Конструкции типа «мне нужен»

```
    "AgentAdjNeed": {
        "example": "мне нужен",
        "head": {
            "tag": "ADJS,Qual"
        },
        "subord": {
            "tag": "NPRO,datv",
        },
```

```
        "target": {
            "copy": { "Head": True },
            "pos": "Left",
        }
    },
```

16. Присоединение дополнения к конструкциям типа «мне нужен»

```
    "AdjNeedObject": {
        "head": {
            "tag": "ADJS,Qual"
        },
        "subord": {
            "tag": "NP, NMbr, accs",
        },
        "target": {
            "copy": { "Head": True },
            "pos": "Right",
        }
    },
```

17. Согласование имени и фамилии

```
    "FullName": {
        "example": "Владимир Путин, Ангела Меркель",
        "head": {
            "tag": "NP, Name"
        },
        "subord": {
            "tag": "NP, Surn, GNdr, NMbr, CAse",
```

```
        },
        "target": {
            "copy": { "Head": True },
            "pos": "Right",
        }
    },
```

18. Вводное слово «пожалуйста»

```
    "Please": {
        "example": "пожалуйста покажи",
        "head": {
            "tag": "VP, impr|NP|PP"
        },
        "subord": {
            "tag": "INTJ",
        },
        "target": {
            "copy": { "Head": True },
            "pos": "Both",
        }
    },
```

19. Обращение

```
    "Address": {
        "example": "яндекс, покажи",
        "head": {
            "tag": "VP, impr|NP|PP"
        },
        "subord": {
```

```
            "tag": "NP, nomn, Name|NP, nomn, Orgn",
        },
        "target": {
            "copy": { "Head": True },
            "pos": "Both",
        }
    },
```

20.Присоединение токенов-чисел

```
"Number": {
        "head": {
            "tag": "VP|NP|PP"
        },
        "subord": {
            "tag": "D",
        },
        "target": {
            "copy": { "Head": True },
            "pos": "Right",
        }
    },
```